

## **BAB II**

### **TINJAUAN PUSTAKA DAN DASAR TEORI**

#### **2.1 Tinjauan Pustaka**

Pada tinjauan pustaka, peneliti mengawali dengan menelaah penelitian terdahulu yang memiliki keterkaitan serta relevansi dengan penelitian yang dilakukan. Dengan demikian, peneliti mendapatkan rujukan pendukung, pelengkap serta pembanding yang memadai sehingga penulisan penelitian ini lebih memadai. Hal ini dimaksudkan untuk memperkuat kajian pustaka dengan didukung penelitian yang ada. Selain itu, karena pendekatan penelitian ini menghargai berbagai perbedaan yang ada serta cara pandang mengenai objek-objek tertentu, sehingga meskipun terdapat kesamaan maupun perbedaan adalah suatu hal yang wajar dan dapat disinergikan untuk saling melengkapi.

Menurut Sandi Pramono Adi (2013), dengan judul “Perancangan Sistem Aplikasi Penjualan Toko Komputer Mascom Berbasis Desktop”, Sistem ini menggunakan metode *SDLC*. Metode *SDLC* adalah metode dengan proses pengembangan dan perancangan sistem hingga pengoperasian dan pemeliharaan. Penyimpanan data menggunakan database jenis *MySQL*. Perancangan sistem menggunakan aplikasi *Netbeans IDE* dengan seperangkat *JDK*, dan untuk pembuatan laporan atau nota menggunakan aplikasi *iReport*. Sistem Aplikasi Penjualan Toko MasCom ini memiliki fasilitas pengolahan data pembelian, penjualan, pemesanan barang, dan penyetokan barang beserta data laporan. Sehingga adanya sistem baru tersebut diperoleh jumlah sebanyak 83,3% peserta yang menyatakan keseluruhan sistem layak dan baik untuk digunakan dalam membantu toko MasCom untuk mengolah data, pembuatan laporan, serta

memudahkan dalam mendapatkan informasi tentang data yang diolah baik untuk toko MasCom itu sendiri.

Menurut Dani Setiadi (2013), dengan judul “Perancangan Sistem Informasi Penjualan Dan Pembelian Obat Di Apotek Emulinda Bandung”, selain dalam proses perhitungan masih menggunakan alat elektronik kalkulator dimana prosesnya harus menjumlahkan satu persatu harga obat yang harus di bayarkan oleh konsumen. Hal ini yang menjadi salah satu kendala yang ada di Apotek Emulinda karena ada beberapa transaksi terjadi kekeliruan atau kesalahan di dalam proses perhitungan pembayaran oleh bagian kasir di saat apotek ramai di datangi Konsumen. Dengan adanya kekeliruan atau kesalahan di dalam proses perhitungan pembayaran maka menyebabkan terjadinya sehingga dapat menimbulkan peluang kesalahan dalam pencatatan dan pembuatan laporan data transaksi yang tidak sesuai dengan data obat yang keluar.

Menurut Kristianto (2012), dengan judul “Perancangan Sistem Informasi Kasir Pada Toko “Rejomaret” Di Wonogiri”, Sistem informasi Kasir ini dapat menjadi salah satu solusi yang dapat digunakan untuk mempermudah pekerja dalam mengolah data penjualan barang, serta membuat laporan penjualan barang dengan lebih mudah, cepat dan efisien, dan hasil pengolahan data pada sistem informasi Kasir berupa laporan-laporan, yaitu laporan data barang, data supplier, data Kategori, penjualan per Tanggal, penjualan per periode, penjualan bulanan, penjualan mingguan, penjualan harian, pembelian per periode, pembelian mingguan, pembelian bulanan dan seluruh pembelian.

Menurut Novita Rini (2010), dengan judul “Sistem Informasi Penjualan Barang Toko Sumber Urip”, Sistem Informasi Penjualan Barang adalah sistem informasi yang menyediakan informasi dari beberapa proses yang meliputi penjualan (Accounting/manajemen), stock (pengadaan barang), pergudangan (inventory), dan pelaporan. Tujuan penelitian ini adalah untuk membuat Sistem Informasi Penjualan Barang di Toko Sumber Urip. Metode yang digunakan adalah metode observasi, metode wawancara, dan metode studi literature. Software pendukung dalam pembuatan aplikasi ini adalah macromedia dreamweaver 8, PHPMYAdmin, MySQL.

Tabel 2.1 Tinjauan Pustaka

NO	Parameter / Tahun	Judul	Bahasa Pemrograman	scan barcode	Platform
1	Sandi Pramono Adi (2013)	Perancangan Sistem Aplikasi Penjualan Toko Komputer Mascom Berbasis Desktop	Java Dekstop	Ya	Dekstop
2	Dani Setiadi (2013)	Perancangan Sistem Informasi Penjualan Dan Pembelian Obat Di Apotek Emulinda Bandung	Microsoft Visual Basic 6.0	Ya	Dekstop
3	Kristianto (2012)	Perancangan Sistem Informasi Kasir Pada Toko “Rejomaret” Di Wonogiri	Microsoft Visual Basic 6.0	Ya	Dekstop
4	Novita Rini (2010)	Sistem Informasi Penjualan Barang Toko Sumber Urip	PHP	Tidak	Dekstop

5	Usulan Peneliti (2020)	Rancang Bangun Sistem Point Of Sales Dengan Metode System Development Life Cycle (SDLC) Pada Toko Total Sport Berbasis Web	PHP	Ya	Dekstop
---	------------------------	--	-----	----	---------

### 2.1.1 Persamaan dan Perbedaan Antara Penelitian Sebelumnya

Pada penelitian di atas, terdapat kesamaan dan perbedaan dari penelitian yang akan dibuat saat ini. Adapun persamaan sistem di atas ialah pada pemanfaatan sistem pengolahan data penjualan, pembelian, penyetokan barang serta data laporan. Namun, yang menjadi perbedaan dengan sistem yang dibuat oleh Sandi Pramono Adi adalah focus dalam memudahkan saat pelaporan sedangkan peneliti focus dengan pelayanan dalam bertransaksi.

Pada penelitian di atas, terdapat kesamaan dan perbedaan dari penelitian yang akan dibuat saat ini. Adapun persamaan sistem di atas ialah pada pemanfaatan sistem dalam mengolah data penjualan. Namun, yang menjadi perbedaan dengan sistem yang dibuat oleh Dani Setiadi adalah jenis pemasaran produknya di mana penelitian di atas menggunakan sistem web penjualan, sedangkan peneliti menggunakan jenis web *Point of Sale*.

Pada penelitian di atas, terdapat kesamaan dan perbedaan dari penelitian yang akan dibuat saat ini. Adapun persamaan sistem di atas ialah pada pemanfaatan sistem dalam mengolah data penjualan. Namun, yang menjadi

perbedaan dengan sistem yang dibuat oleh Kristianto adalah sistemnya berfokus dalam rincian laporan sedangkan peneliti berfokus dalam kecepatan bertransaksi.

Pada penelitian di atas, terdapat kesamaan dan perbedaan dari penelitian yang akan dibuat saat ini. adapun persamaan sistem di atas ialah pada pemanfaatan sistem dalam informasi penjualan. Namun, yang menjadi perbedaan dengan sistem yang dibuat oleh Novita Rini adalah sistemnya berfokus pemasaran informasi produknya berbasis online sedangkan peneliti sistem pemasaran berfokus pada transaksi langsung di toko.

## **2.2 Dasar Teori**

### ***2.2.1 Software Development Life Cycle (SDLC)***

Menurut Sukamto dan Shalahuddin (2013:26) mengemukakan bahwa “ SDLC atau *Software Development Life Cycle* atau sering disebut juga *System Development Life Cycle* adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya, berdasarkan *best practice* atau cara-cara yang teruji baik. ”

*Software Development Life Cycle (SDLC)* atau disebut juga dengan siklus hidup pengembangan perangkat lunak adalah sebuah proses yang digunakan untuk merancang, mengembangkan, dan menguji perangkat lunak berkualitas tinggi. SDLC bertujuan untuk menghasilkan perangkat lunak berkualitas tinggi yang memenuhi, bahkan melampaui harapan pelanggan, dengan penyelesaian yang tepat waktu dan sesuai dengan rencana anggaran.

- SDLC adalah akronim dari *Software Development Life Cycle*, Siklus Hidup Pengembangan Perangkat Lunak.
- SDLC biasa juga disebut dengan *Software Development Process*, Proses Pengembangan Perangkat Lunak.
- SDLC adalah kerangka kerja yang mendefinisikan pekerjaan yang dilakukan pada setiap langkah dalam proses pengembangan perangkat lunak.
- ISO / IEC 12207 adalah standar internasional untuk proses siklus hidup perangkat lunak, bertujuan menjadi standar yang mendefinisikan semua tugas yang diperlukan untuk mengembangkan dan memelihara perangkat lunak.

### **2.2.2 Tahapan *Software Development Life Cycle***

Siklus Hidup Pengembangan Perangkat Lunak biasanya terdiri dari tahapan-tahapan berikut:

#### **Tahap 1: Analisis Perencanaan dan Kebutuhan (*Planning and Requirement Analysis*)**

Analisis kebutuhan adalah tahap paling penting dan mendasar dalam *SDLC*. Biasanya, analisis ini dilakukan oleh anggota tim senior. Masukan dari customer, juga divisi lain, seperti *sales*, *market research*, dan pakar-pakar lainnya sangat dibutuhkan. Informasi ini kemudian digunakan untuk melakukan pendekatan dasar project, dan studi kelayakan dari segi keuangan, operasional dan teknis.

Persyaratan yang diperlukan dalam melakukan *quality assurance*, juga direncanakan pada tahap ini. Sehingga, bisa menghasilkan suatu studi kelayakan teknis untuk mendefinisikan berbagai pendekatan teknis yang dapat diikuti dengan sukses dan minim risiko.

### **Tahap 2: Menentukan Persyaratan (*Defining Requirements*)**

Mendefinisikan dan mendokumentasikan persyaratan produk dengan jelas dan rinci untuk disetujui customer dan atau *market analyst*, merupakan langkah berikutnya. Spesifikasi Kebutuhan Perangkat Lunak (*Software Requirement Specification/SRS*) merupakan nama dokumen dalam tahap ini. Berisikan semua persyaratan produk yang akan dirancang dan dikembangkan selama siklus hidup proyek.

### **Tahap 3: Merancang Arsitektur Produk (*Designing Product Architecture*)**

SRS yang dihasilkan pada tahap sebelumnya, merupakan referensi utama *product architects* untuk membuat rancangan software. Biasanya, akan ada lebih dari satu rancangan dan pendekatan design yang dihasilkan dan dikomumentasikan dalam Spesifikasi Dokumen Desain (*Design Document Specification/DDS*).

*DDS* ini lah yang ditinjau oleh semua pemangku kepentingan. Desain yang dipilih biasanya diambil berdasarkan berbagai pertimbangan dan parameter, seperti, risiko, ketahanan produk, modularitas desain, anggaran, dan batasan waktu. Tentu, desain dengan pendekatan terbaik terhadap parameterlah yang dipilih.

Pendekatan desain harus dengan jelas mendefinisikan semua modul arsitektur produk bersama dengan aliran data dan komunikasi dengan pihak ketiga dan eksternal (jika ada). Desain internal semua modul arsitektur yang diusulkan harus didefinisikan secara jelas hingga rincian terkecil dalam *DDS*.

#### **Tahap 4: Membangun atau Mengembangkan Produk (*Building or Developing Product*)**

Pada tahap ini lah, produk sebenarnya dibangun. *DDS*, menjadi dasar tiap *coding* yang dibangun. Idealnya, jika desain dan perencanaan dilakukan secara terperinci dan terorganisir, pembuatan kode dapat dilakukan tanpa banyak kesulitan.

Pengembang (*developers*) harus mengikuti pedoman pengkodean (*coding guidelines*) yang ditentukan oleh organisasi seperti compiler, interpreter, debugger, dll yang digunakan untuk menghasilkan kode. Bahasa pemrograman seperti *C*, *C ++*, *Pascal*, *Java* dan *PHP* digunakan dalam pemrograman ini. Bahasa dipikh dehubungan dengan jenis *software* yang dikembangkan.

#### **Tahap 5: Menguji Produk (*Testing the Product*)**

Dalam model SDLC modern, biasanya, pengujian dilakukan di tiap tahapan. Namun, yang dimaksud pada pengujian dalam tahap ini ialah tahap pengujian di mana cacat produk dilaporkan, dilacak, diperbaiki dan diuji ulang, sampai produk mencapai standar kualitas yang ditentukan dalam SRS.

### **Tahap 6: Penempatan di Pasar dan Pemeliharaan (*Deployment in the Market and Maintenance*)**

Setelah produk diuji dan siap digunakan, produk akan dirilis secara resmi di pasar yang sesuai. Bisa secara bertahap maupun keseluruhan, berdasarkan strategi bisnis dan organisasi yang bersangkutan. Biasanya, produk pertama kali dirilis pada segmen terbatas dalam lingkup bisnis yang nyata. (uji penerimaan pasar, *User Acceptance testing/UAT*).

Kemudian, berdasarkan umpan balik, produk dirilis kembali dengan penambahan yang disarankan pasar. Pemeliharaan pun, dilakukan untuk basis pelanggan yang ada.

#### **2.2.3 Model-Model SDLC**

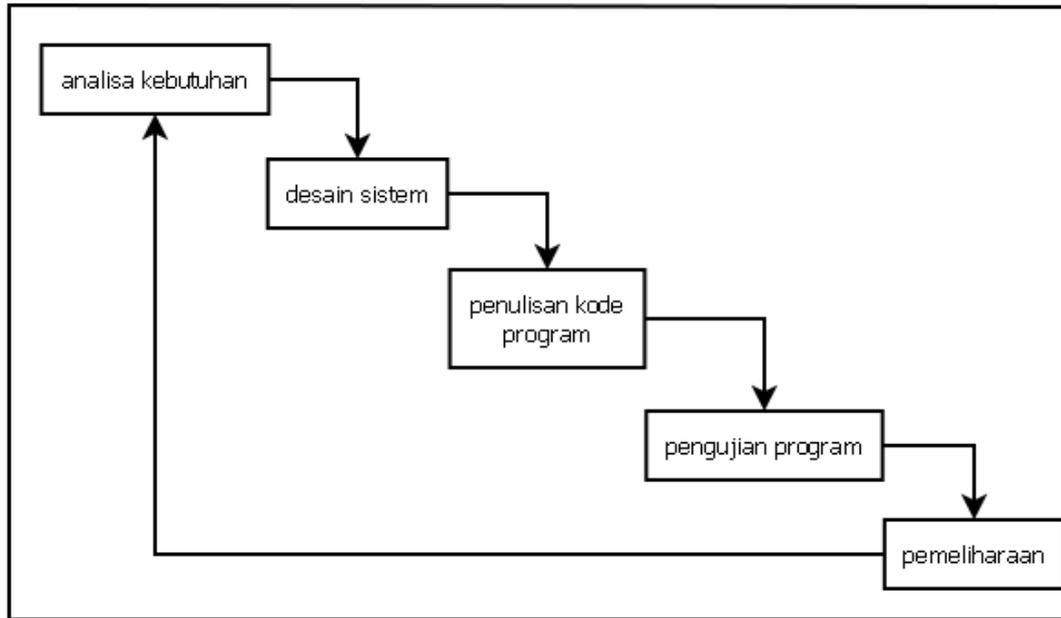
Ada berbagai model siklus hidup pengembangan perangkat lunak. Model-model ini juga disebut sebagai *Software Development Process Models*. Setiap model proses mengikuti serangkaian langkah-langkah unik untuk memastikan keberhasilan dalam proses pengembangan perangkat lunak.

Berikut ini adalah model SDLC yang paling penting dan populer:

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model

Beberapa metode lainnya ialah Agile Model, RAD Model, Rapid Application Development and Prototyping Models.

Pada penelitian ini akan dikembangkan dengan waterfall Model, Berikut ini merupakan tahapan tahapan pengembangan dalam metode *waterfall*.



Gambar 2.1 Tahap pemodelan *waterfall*

### 1. Analisis Kebutuhan

Analisis kebutuhan sistem terdapat dua hal yang akan dibahas yakni kebutuhan fungsional dan kebutuhan non fungsional. Kedua kebutuhan ini dapat disimpulkan dari desain sistem yang diteliti. Yaitu:

#### 1) Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan fungsi atau fitur apa saja yang terdapat didalam sistem, fitur-fitur tersebut meliputi masukan (*input*), proses dan keluaran (*output*). Beberapa Kebutuhan fungsionalitas sistem *point of sale* yang telah

disimpulkan adalah sebagai berikut.

Tabel 2.2 Kebutuhan fungsional

<b>Kebutuhan Fungsional</b>	<b>Deskripsi</b>
Kelola Barang	<i>Administrator</i> akan mengelola data input barang persediaan serta mengubah data dalam sistem.
Kelola Pembelian	<i>Administrtor</i> akan mengelola data pembelian barang didalam sistem, kelola pembelian ini meliputi inputan transaksi pembelian barang pengubahan data pembelian dan penghapusan data beli.
Kelola Mutasi Barang	<i>Administrator</i> adalah pihak yang mengelola mutasi barang di dalam sistem, pengelolaan mutasi barang meliputi penambahan data mutasi barang, pengubahan data dan penghapusan data mutasi barang.
Transaksi	<i>Administrator</i> dan Kasir adalah pihak yang dapat menggunakan fitur tersebut namun pihak kasir hanya dapat memasukkan dua tipe transaksi yaitu transaksi utama dan jasa <i>service</i> sedangkan untuk <i>administrator</i> beberapa transaksi penjualan dan transaksi pembelian dapat digunakan. Untuk beberapa tranasaksi tersebut pihak <i>administrator</i> bertanggung jawab penuh mulai dari penginputan, penghapusan yang berkaitan dengan transaksi.
Laporan	<i>Administrator</i> dan <i>manager</i> adalah pihak yang dapat mengakses kebutuhan pelaporan.
Kelola Admin	<i>Administrator</i> adalah pihak yang berkuasa penuh dalam mengelola Admin atau mengelola pihak mana yang dapat mengakses sistem. Pihak <i>administrator</i> dapat menambahkan pegawai ataupun menghapus pegawai yang sudah tidak bekerja lagi.

Data Request	<i>Manager</i> dan kasir adalah pihak yang berhubungan dengan data request melalui sistem tersebut.
Sinkronasi Data	<i>Administrator</i> dan Kasir merupakan pihak yang dapat memperbarui data yang akan dikirim ke <i>database master POS</i> serta database klien penjualan dan sebaliknya.

## 2) Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional sistem merupakan analisis kebutuhan yang menentukan kriteria yang dapat digunakan untuk menilai fungsionalitas dari sistem. Beberapa fungsionalitas dari kebutuhan non fungsional melibatkan analisis kebutuhan terhadap pengguna (*user*) dan sistem. Kebutuhan non fungsional pada sistem ini dibagi menjadi 2 yaitu :

### 1. Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna ini adalah analisis terhadap kebutuhan pengguna yang langsung berinteraksi dengan sistem. Beberapa pengguna yang langsung berinteraksi dengan sistem *point of sale* adalah seluruh pegawai dan pemilik toko retail Total Sport. untuk level tertinggi didalam sistem tersebut yaitu admin toko sebagai *administrator*, pemilik toko sebagai *manager*, pegawai kasir sebagai pihak yang melakukan sub sistem *point of sale*. Penjelasan untuk setiap pengguna dapat dilihat pada Tabel 3..

Tabel 2.3 Analisis Kebutuhan Pengguna

<b>Pengguna</b>	<b>Deskripsi</b>
Admin toko ( Administrator )	Pengguna merupakan beberapa pegawai yang menjadi tangan kanan pemilik toko JSC Jasa Sarana Computer dalam mengelola Sistem
Pemilik toko ( Manager )	Pengguna merupakan pemilik toko JSC
Kasir	Pengguna merupakan pegawai kasir yang bertugas untuk melayani pelanggan dalam Bertransaksi

## 2. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem merupakan analisis sebagai acuan yang ditujukan ke sistem untuk menjadi pendukung pada fungsional sistem. Berikut beberapa penjelasan analisis kebutuhan sistem yang ada di sistem *point of sale* akan dapat dilihat pada

Tabel 3.6

Tabel 2.4 Analisis Kebutuhan Sistem

<b>Kebutuhan</b>	<b>Deskripsi</b>
Hak Akses <i>Level</i>	Sistem <i>Administrator, manager</i> dan pegawai/kasir. Masing-masing mempunyai hak akses dan fungsional yang berbeda.
Print laporan	Sistem dapat melakukan cetak laporan transaksi atau struk untuk pembeli.
<i>Auto Suggestion</i>	Sistem penjualan ini memakai <i>auto suggestion</i> untuk memudahkan user dalam pencarian barang.

<i>Auto Report</i>	Sistem ini dapat melakukan olah laporan secara otomatis sesuai laporan yang diinginkan.
Aksesibilitas	Sistem dapat digunakan hanya dilingkungan Toko <i>JSC</i> Jasa Sarana <i>Computer</i> sedangkan untuk <i>manager</i> mendapatkan aplikasi <i>mobile</i> yang dapat diakses dimanapun asalkan tersambung oleh jaringan internet.
Performa	Sistem dapat mengirimkan beberapa data dengan mengimplmentasikan <i>web service</i> dari <i>database POS</i> ke <i>database</i> master untuk pembaharuan data yang akan diakses melalui <i>mobile</i> .

## **2. Desain Sistem**

Desain perangkat lunak merupakan proses multi langkah yang focus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat *lunak*, representasi antar muka, dan prosedur pengkodean. Sistem ini dapat melakukan bermacam keperluan dalam mempermudah pekerjaan user seperti *input* data transaksi dengan *output* nota bagi pelanggan, melakukan request barang, mengolah data transaksi dengan *output* pelaporan, serta keperluan manajemen untuk pengontrolan data-data didalam sistem. Untuk struktur data dapat dilihat di data *flow diagram* yang secara garis besar menunjukan cara kerja dari setiap *user*.

## **3. Pengkodean**

Pengkodean merupakan desain yang di translasikan ke dalam *program perangkat lunak*. Hasil dari tahap ini adalah *program komputer* sesuai dengan desain yang telah dibuat pada tahap desain. Dari desain yang

telah digambarkan tahapan selanjutnya adalah merealisasikan nya kedalam bahasa pemrograman, dengan melakukan *pengkodean* ini diharapkan dapat mengatasi kendala dalam melakukan pencarian informasi seperti pengaduan pelanggan, monitoring penanganan, dan kesalahan penghitungan pembayaran, serta kerumitan dalam menyusun laporan bulanan seperti laporan pengaduan maupun laporan penanganan.

#### **4. *Testing / Pengujian***

Testing/pengujian focus pada perangkat lunak secara dari segi logic dan fungsional dan memastikan bahwa semua bagian sudah di uji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang di hasilkan sesuai dengan yang diinginkan.

#### **5. *Maintenance***

Melakukan pemeliharaan terhadap perangkat *lunak* yang telah di rancang. Yaitu baik dengan perawatan secara terjadwal rutin untuk pengecekan sistem ataupun dengan pemeliharaan dari unsur eksternal dari pemakai sistem.

### **2.2.4 *Point Of Sale***

Menurut Widjanto (2001: 343), “Sistem *Point-of-sale (POS)* merupakan sistem pengolahan data yang banyak digunakan pada bisnis pengecer (*retail*), seperti di pasar swalayan, restoran siap-saji, ataupun pusat-pusat perbelanjaan. Di perusahaan-perusahaan tersebut, mesin *cash register* pada umumnya memiliki kemampuan yang lebih canggih dan berfungsi sekaligus sebagai terminal.” Suatu

alat yang sangat besar perannya dalam sistem *POS* adalah *POS recorder*. Alat ini bisa membaca harga atau data kode barang, dan biasanya ditempatkan pada lokasi kasir.

*Point of Sales* atau disingkat sebagai *POS*, secara sederhananya dapat diartikan sebagai sebuah perangkat lunak yang melakukan pencatatan akan transaksi jual beli yang terjadi pada perusahaan (ST, 2010). Sebelum sistem *POS* dikembangkan, orang-orang banyak menggunakan yang kita kenal sebagai mesin *Cash Register* atau bahkan pencatatan manual menggunakan nota dalam menjalankan usahanya. Seiring dengan kemajuan teknologi, *Cash Register* ini dianggap sudah tidak cukup memadai lagi, karena sistem *Cash Register* hanya dapat mencatat/menghitung transaksi. Sehingga muncul pengembangan baru yaitu perangkat lunak *POS*. Beberapa kegunaan system *POS* adalah sebagai berikut:

1. Peningkatan kualitas pelayanan
2. Peningkatan citra usaha
3. *Competitive advantage*
4. Kemudahan proses control & pengambilan keputusan.

Point of sales atau disingkat *POS* secara umum dapat diartikan sebagai sebuah sistem yang memungkinkan diadakannya proses transaksi. *POS* dapat digunakan di semua transaksi penjualan seperti restoran, supermarket, hotel, dan toko-toko retail. Karena itu, *POS* juga dapat diartikan sebagai proses pelayanan transaksi dalam sebuah toko retail. Dari semua pengertian yang dijelaskan tersebut, maka dapat diambil kesimpulan bahwa *point of sales* dapatdiartikan sebagai sebuah sistem yang memungkinkan diadakannya transaksi yang

didalamnya termasuk juga penggunaan mesin kasir.

Menurut Rokhman (2012), perangkat lunak *point of sales (POS)* adalah perangkat lunak yang banyak digunakan pada usaha retail seperti swalayan, mini market, apotek, cafe, dan lain-lain. Secara umum proses-proses yang biasanya digunakan di setiap sistem *point of sales* yang terdapat di perusahaan-perusahaan adalah sebagai berikut :

1. *Point of sales* (transaksi penjualan)
2. *Inventory control* (Pengendalian persediaan barang)
3. Pembacaan *barcode*
4. Manajemen toko
5. Retur Penjualan
6. Pelaporan

Seiring dengan kemajuan dalam dunia teknologi informasi, penggunaan *Hardware* seperti komputer dengan berbagai aplikasi yang mendukung kinerja komputer. Ini juga berpengaruh pada bisnis retail yang secara fungsinya mesin kasir atau cash register sebagai media *point of sales*. Dalam perkembangannya, mesin kasir yang menjadi *point of sales* ini sudah dilengkapi dengan aplikasi *point of sales* yang bertujuan mempermudah dan mempercepat transaksi jual beli dengan sistem yang mudah untuk digunakan.

### **2.2.5 Database**

Basis data (*database*) adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program

komputer untuk memperoleh informasi dari basis data tersebut.

Menurut Andino Maselena (2008:47), “*Database* adalah sekumpulan *file* yang saling terkait dan membentuk suatu bangun data. *Database* minimal terdiri dari satu *file* yang cukup untuk dimanipulasi oleh komputer sedemikian rupa.”

*Database* dapat diartikan sebagai himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

Dari uraian di atas, *Database* dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan, yang dikelola dan disimpan secara terintegrasi melalui metode tertentu dengan menggunakan komputer.

### **2.2.6 Website**

Menurut Sibero (2014:11), “ *Website (world wide web /W3)* adalah suatu sistem yang berkaitan dengan dokumen digunakan sebagai media untuk menampilkan teks, gambar, multimedia, dan lainnya pada jaringan *internet*.

Menurut Kustiyahningsih dan Rosa Anamisa (2011b:4) “*website* merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet *hypertext* untuk menampilkan data berupa teks, gambar, suara, animasi dan data multimedia lainnya”.

Kesimpulannya, *website* adalah sebuah informasi melalui internet, yang merupakan kumpulan-kumpulan dokumen yang banyak tersebar di beberapa server yang ada di penjuru dunia dan terhubung menjadi satu jaringan melalui jaringan yang disebut internet. Situs web biasanya ditempatkan setidaknya pada

sebuah server web yang dapat diakses melalui jaringan seperti *internet*, atau pun jaringan wilayah lokal (*LAN*) melalui alamat *internet*, yang dikenali sebagai *URL*. Gabungan atas semua situs yang dapat diakses publik di *internet* disebut pula sebagai *World Wide Web* atau lebih dikenal dengan singkatan *WWW*.

### 2.2.7 Bahasa Pemrograman

#### a. Pengenalan *PHP*

*PHP* secara resmi merupakan kependekan dari *PHP:HyperText Preprocessor*, yang merupakan bahasa *script server-side* yang bersifat *open source*. Bahasa *PHP* menyatu dengan *script HTML* yang sepenuhnya dijalankan pada server.

Menurut Anhar (2010:7), “*PHP:HyperText Preprocessor* adalah bahasa *interpreter* yang banyak digunakan dalam internet dan intranet, sebagai program yang diintegrasikan ke dalam *web server* seperti *Apache* dan *Microsoft Internet Information Service (IIS)*. Karena *PHP* ini dieksekusi di server, maka *PHP* disebut sebagai *Server Side Scripting Language*.”.

Beberapa hal yang menjadi keunggulan *PHP*, antara lain:

- a. Mempunyai dukungan kepada banyak database.
- b. Fleksibilitas yang tinggi dan memiliki banyak fungsi *Built-in*.
- c. Mendukung pemrograman berorientasi objek (*OOP*).
- d. Berjalan di banyak *platform* seperti *Windows*, *LINUX* dan *MacOS*.
- e. *PHP* adalah produk *open source* yang gratis (*free*), sehingga dapat didistribusikan tanpa dikenakan biaya.

Cara penulisan skrip *PHP* ada dua macam, yaitu *singlefile* dan *embedded*.

Contoh :

```
Singlefile
echo "<html>";
echo "<head>";
echo "<title>contoh php</title> ";
echo "</head>";
echo "</body>";

echo "<p> dibawah ini adalah tulisan hasil dari
PHP</p>";
echo "teks dari PHP";
echo "</html>";

Embedded
<head>

<title> contoh PHP</title>

</head>

</body>

<p> dibawah ini adaah tulisan dari PHP</p>

<?PHP
        Echo "teks dari PHP";

?>

</body>

</html>
```

#### b. HTML (*Hypertext Markup Language*)

Menurut Sibero (2014:19) "*HTML (Hyper Text Markup Language)* adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen web". Struktur dokumen *HTML* terdiri dari *tag* pembuka dan *tag* penutup.

Dalam penulisan struktur Dokumen *HTML* terdapat 3 hal yang harus di perhatikan :

### 1) *HTML*

Setiap *HTML* diawali dan ditutup dengan *tag HTML*.dokumen

### 2) *HEAD*

Bagian *head* biasanya berisi *tag TITLE*, *Meta tag*, dan semua *script* java atau yang lainnya yang di eksekusi di *browser*. Di bagian inilah kita memberikan *bookmark* untuk keperluan pencarian (*searching*).

### 3) *BODY*

Pada bagian ini digunakan untuk menampilkan *text*, *image*, *link*, dan semua yang ditampilkan pada *web page*.

Menurut Kustiyaningsih dan Anamisa (2011c:13) “*HTML* kependekan dari *Hyper Text Markup Language*. Dokumen *HTML* adalah *file text* murni yang dapat dibuat dengan editor teks sembarang”. Dokumen ini dikenal sebagai *web page*. File-file *HTML* ini berisi instruksi-instruksi yang kemudian diterjemahkan oleh browser yang ada dikomputer *client (user)* sehingga isi informasinya dapat ditampilkan secara visual di computer pengguna (*user*).

#### c. *CSS (Cascading Style Sheet)*

Menurut Supardi (2010:128) mengemukakan bahwa “*CSS* merupakan format atau style untuk tampilan web, dapat dibuat dalam halaman bersama perintah *HTML* atau file tersendiri dengan nama perluasan *css*”.

*CSS* dikembangkan untuk menata gaya pengaturan haman *web*. Pada walanya *CSS* dikembangkan pada *SGML* pada tahun 1970. Pada bulan Desember 1996 *W3C* mengenalkan spesifikasi *CSS level 1* atau dikenal sebagai *CSS!* Yang mendukung format huruf, warna pada teks, dan lain-lain. Pada bulan Mei 1998

W3C mengeluarkan CSS2 yang didalamnya terdapat fungsi pengaturan letak elemen, dan versi terbaru CSS yang mampu melakukan desain *website*, diantaranya animasi warna hingga animasi 3D. Saat ini CSS telah mendukung banyak bahasa seperti *HTML*, *XHTML*, *XML*, *SVG (Scalable Vector Graphics)*, dan *Mozilla XUL (XML User Interface Language)*. Berikut contoh sederhana penulisan CSS :

```
<div style = "font-weight:bold">
<p>Contoh paragraph dalam DIV dengan style
font:bold</p>

<span> Contoh span dalam DIV dengan
style
font- weight:bold</span>
</div>
```

#### d. Pengenalan Basis Data *MySQL*

*MySQL* adalah *Multiuser Database* yang menggunakan bahasa *Structured Query Language (SQL)*. *MySQL* merupakan *software Database Management System (DBMS)* yang sangat populer di kalangan pemrogram *web*. *MySQL* pertama dibuat oleh *Michael Monty Widenius* dan kawan-kawannya pada tahun 1994.

*MySQL* memiliki beberapa keistimewaan, antara lain:

- a. Portabilitas. *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *UNIX*, *MacOS* dan masih banyak lagi.
- b. Perangkat Lunak Sumber Terbuka (*Open Source*). *MySQL* didistribusikan sebagai perangkat lunak sumber terbuka, sehingga dapat digunakan secara gratis.
- c. *Multi-user*. *MySQL* dapat digunakan oleh beberapa pengguna dalam

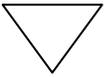
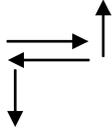
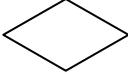
waktu yang bersamaan tanpa mengalami masalah atau konflik.

- d. Tipe Data. *MySQL* memiliki ragam tipe data yang sangat banyak, seperti *signed/unsigned integer*, *float*, *double*, *char*, *varchar*, *text*, *date*, *timestamp* dan lain-lain.
- e. Keamanan. *MySQL* memiliki beberapa lapisan keamanan seperti *level user*, nama host dan izin akses user dengan sistem perizinan yang mendetail serta sandi yang terenkripsi.
- f. Skalabilitas dan Pembatasan. *MySQL* mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 *indeks* pada tiap tabelnya.

### **2.2.8 Flowchart**

Menurut Adelia (2011), *Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong *analyst* dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman.

Tabel 2.5 Simbol-Simbol *Flowchart*

Simbol	Keterangan
	<b>Penghubung</b> Simbol untuk keluar/masuk atau proses dalam lembar atau halaman lain.
	<b>Input Output</b> Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya.
	<b>Dokumen</b> Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau ouput dicetak di kertas.
	<b>On Line Storage</b> Simbol yang menunjukkan bahwa data di dalam simbol ini akan di simpan.
	<b>Simbol Garis Alir</b> Digunakan untuk menunjukkan arah selanjutnya yang akan dituju dari simbol-simbol dan <i>flowchart</i> .
	<b>Simbol Manual</b> Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
	<b>Terminal</b> Simbol yang menunjukkan untuk permulaan atau akhir suatu sistem.
	<b>Kondisi</b> Simbol keputusan yang menunjukkan kondisi.
	<b>Proses</b> Simbol yang menunjukkan pengolahan dilakukan oleh komputer.
	<b>Penghubung</b> Simbol untuk keluar/masuk prosedur atau proses dalam lembar atau halaman yang masih sama.

Sumber : (Jogiyanto, 2000:796)

### 2.2.9 Entity Relationship Diagram (ERD)

Menurut Sutanta (2011:91), “*Entity Relationship Diagram (ERD)* merupakan suatu model data yang dikembangkan berdasarkan objek. *ERD* digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis.” Berikut ini penulis menjabarkan komponen-komponen yang terdapat di dalam *ERD*, yaitu :

1. *Entity*

*Entity* adalah sesuatu yang dapat dibedakan dalam dunia nyata dimana informasi yang berkaitan erat dengannya dikumpulkan. Simbol yang digunakan untuk *entity* adalah persegi panjang

2. *Relationship*

Adalah hubungan yang terjadi antara satu atau lebih *entity*. *Relationship* tidak mempunyai keberadaan fisik, kecuali yang mewarisi hubungan antara *entity* tersebut. Simbol yang di gunakan adalah belah ketupat, *diamond* atau *rectangle*

3. *Atribut*

Atribut adalah karakteristik dari *entity* atau *relationship* yang menyediakan penjelasan detail tentang atau *relationship* tersebut . Gambar atribut diwakili oleh simbol oval.

4. *Cardinality Ratio* atau *Mapping Cardinality*

*Cardinality Ratio* adalah menjelaskan hubungan batasan jumlah keterhubungan satu *entity* dengan *entity* lainnya atau banyaknya *entity* yang berkesesuaian dengan *entity* yang lain melalui *relationship*.

Berikut ini jenis-jenis dari *Cardinaliti ratio* :

a. *One to One* atau Satu ke Satu ( 1:1 )

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada entitas B.

b. *One to Many* (1:M) atau *Many to One* (M:1)

Hubungan satu *entity* dengan banyak *entiy* atau banyak *entity* dengan satu *entity*.

c. *Many to Many* atau Banyak ke Banyak ( M:N )

Hubungan banyak *entity* dengan banyak *entity*.

5. *Derajat Relationship*

Derajat *relationship* adalah menyatakan jumlah *entity* yang berpartisipasi didalam satu *relationship*. Ada 3 jenis derajat *relationship*, yaitu :

a. *Unary degree* (derajat satu) adalah derajat yang memiliki satu *relationship* untuk satu *entity*.

b. *Binary degree* (derajat dua) adalah derajat yang memiliki satu *relationship* untuk dua buah *entity*.

c. *Ternary degree* (derajat tiga) adalah derajat yang memiliki satu *relationship* untuk tiga atau lebih *entity*.

6. *Participation Constraint*

Menjelaskan apakah keberadaan suatu *entity* tergantung pada hubungannya dengan *entity* lain. Di dalam *participation constraint* ini terdapat dua buah *constraint*, yaitu :

*a. Total Participation*

Keberadaan suatu *entity* tergantung pada hubungannya dengan *entity* lain. Di dalam diagram *ERD* di gambarkan dengan dua garis penghubung antar *ERD*

*b. Partial Participation*

Keberadaan suatu *entity* tidak tergantung pada hubungan dengan *entity* lain. Di dalam diagram *ERD* di gambarkan satu garis penghubung.

7. *Logical Record Structure (LRS)*

Menurut Hasugian dan dan Shidiq (2012:608) memberikan batasan bahwa *LRS* adalah “Sebuah model sistem yang digambarkan dengan sebuah *diagram-ER* akan mengikuti pola atau aturan permodelan tertentu dalam kaitannya dengan konvensi ke *LRS*”. *LRS* terdiri dari hubungan diantara tipe *record*. *Link* ini menunjukkan arah dari datu tipe *record* lainnya. Penggambaran *LRS* mulai dengan menggunakan model yang dimengerti. Dua metode yang digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke *LRS*. Metode lain yang dimulai dengan *ERD* dan langsung dikonversikan ke *LRS*.

**2.2.10 Data Flow Diagram (DFD)**

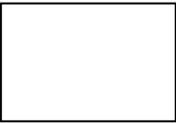
*Data flow diagram* adalah suatu grafik yang menjelaskan sebuah sistem dengan menggunakan bentuk-bentuk dan simbol-simbol untuk menggambarkan aliran data dari proses-proses yang saling berhubungan. *Data flow diagram* ini

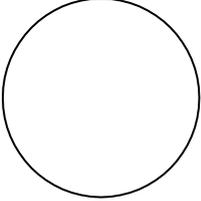
adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem.

Dengan kata lain, *data flow diagram* adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. *Data flow diagram* ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. (Adelia, 2011).

*DFD* sering juga digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem yang baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misal lewat telepon atau surat) serta lingkungan fisik dimana data tersebut akan disimpan (misal *hard disk* atau *disket*). Adapun simbol-simbol yang digunakan dalam *DFD* dapat dilihat pada tabel 2.2.

Tabel 2.6 Simbol *DFD*

Simbol	Keterangan
	<p><b><i>External Entity</i></b> Kesatuan luar merupakan kesatuan (<i>entity</i>) dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada dilingkungan luarnya yang akan memberikan <i>input</i> atau menerima <i>output</i> dari sistem.</p>
	<p><b><i>Data Flow</i></b> Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem</p>

	<p><b><i>Process</i></b> Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin, atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses.</p>
	<p><b><i>Data Store</i></b> Simpanan data merupakan simpanan data suatu <i>file</i>.</p>

Sumber : (Rosa, 2013:71)